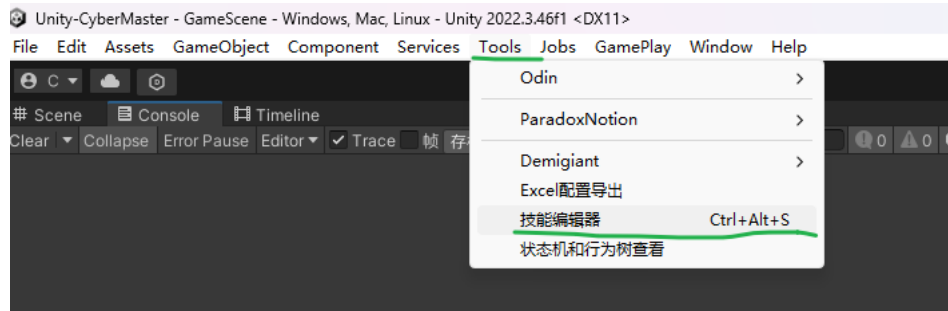


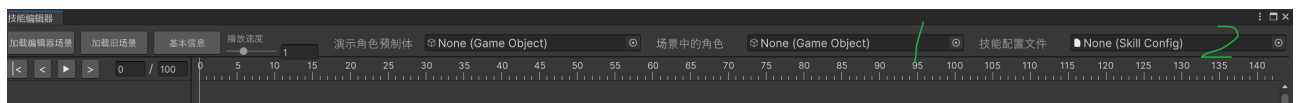
技能编辑器说明

编辑器入口



支持快捷键 ctrl(cmd) + alt + s

打开编辑器后首要关注



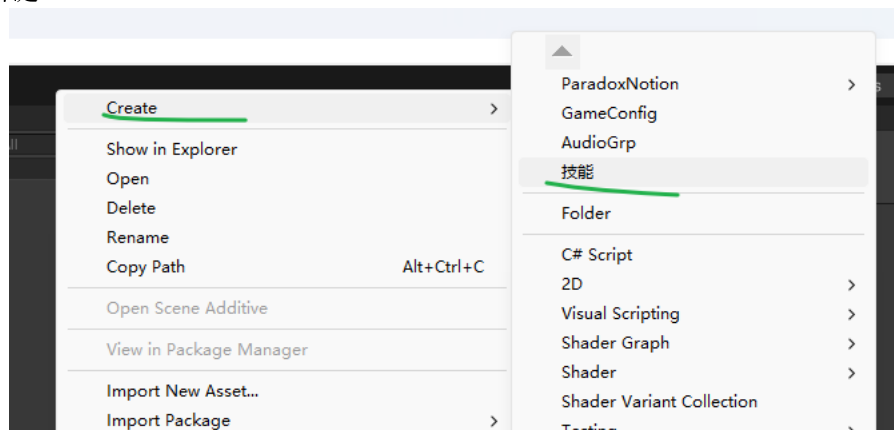
绿 1 需要一个预览用的 GameObject，一般选择挂有 Animator 的物体，比如小兵或者 Boss

绿 2 需要一个技能资源

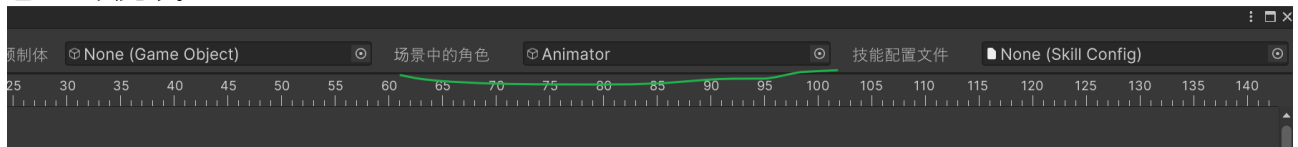
技能现在存放在 Assets/Res/ScriptableObjects/Skill 文件夹里

新建一个技能资源的方式

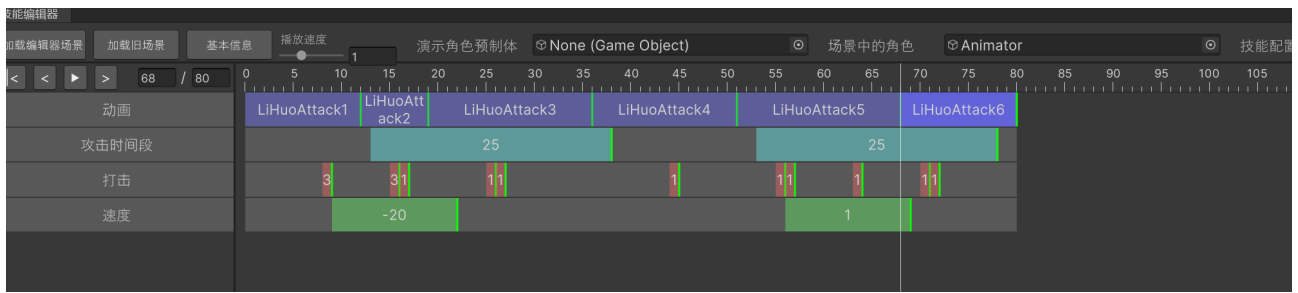
在该文件下按右键



一般推荐先打开一个需要编辑技能的 GameObject 再打开技能编辑器，这样会自动捕获所需的 Animator，比如先打开 Assets\Res\Prefabs\Boss\Boss_LiHuo,再打开技能编辑器，就能发现预览用角色已经填充了。



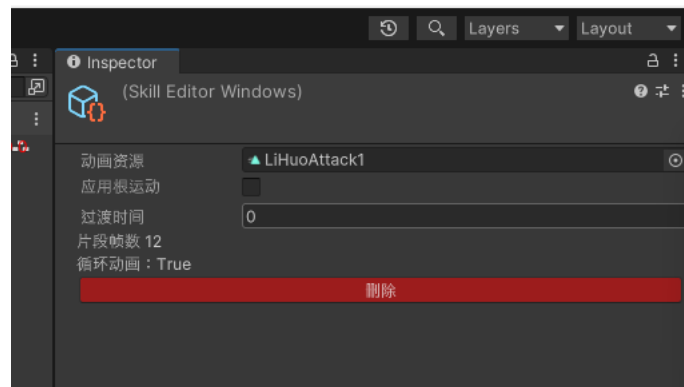
当选取一个技能后能看到技能的类 timeline 结构



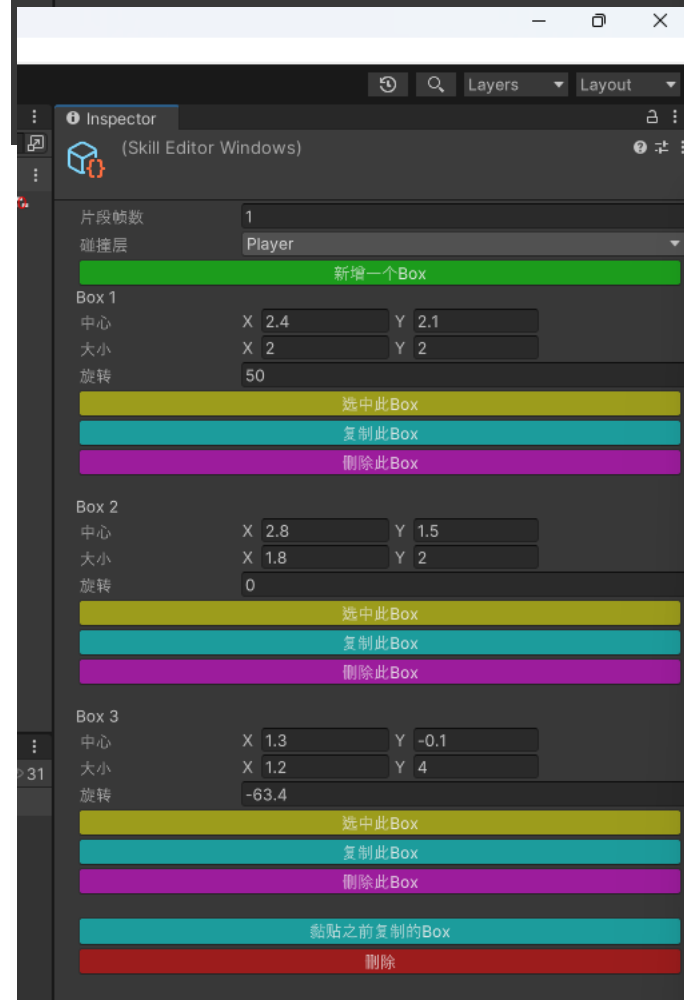
左边是预览播放控制和帧数信息

目前提供了四种轨道，动作，打击 hitbox，攻击时间段，和速度。每一条横行称为一个轨道，轨道上的一小段矩形称为一个 clip，一个 clip 可以包含若干连续的帧，一帧是技能步进的最小时间单位。鼠标单击任意一个 clip 后能在 unity 的 Inspector 窗口看到该 clip 的可编辑内容，不同轨道上的 clip 的可编辑内容不同。按住 clip 后能左右拖动来改变 clip 的起始帧数（目前持续 1 帧的 clip，多见于 hitbox 轨道，会比较难拖动，可以先修改为超过 1 帧后拖动起始帧后在改回 1 帧，未来可能会优化这点），但是目前 clip 之间不能重叠。除了动作 clip 之外，其他的 clip 都能改当前 clip 的持续帧数，动画 clip 的片段帧数由其对应的 AnimationClip 时长决定。

动作 clip



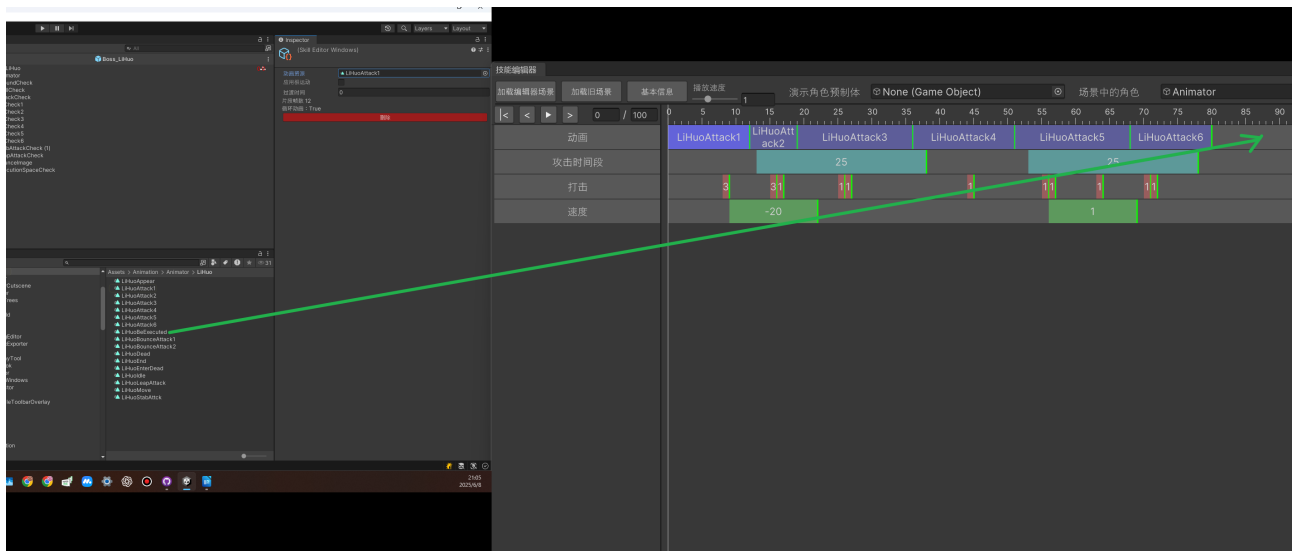
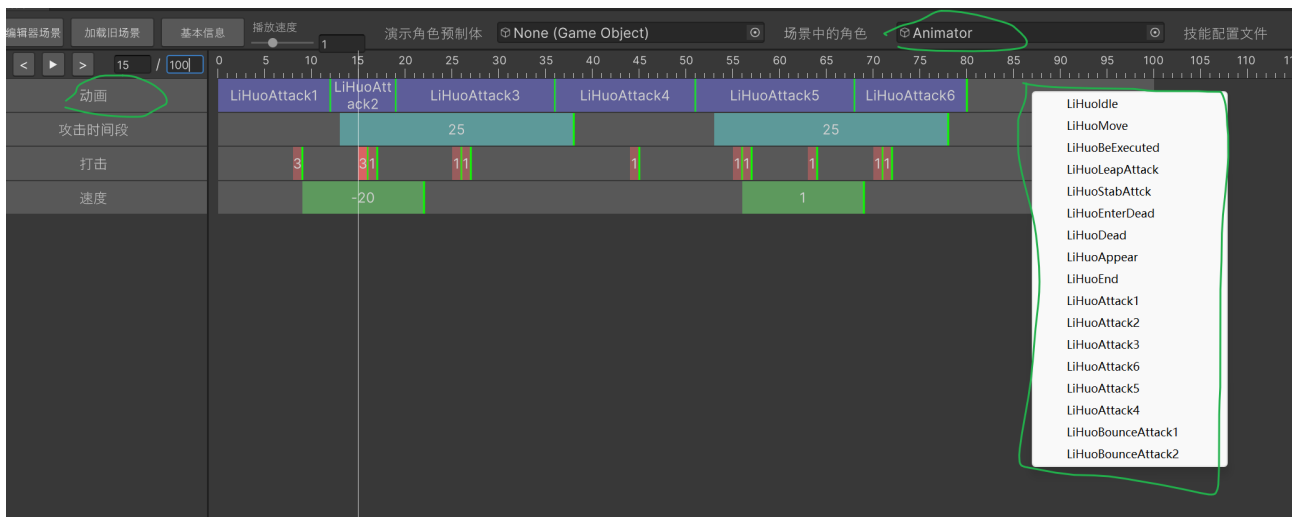
打击 hitbox clip



选中一个 clip 后按 **del** 能删除整个 clip。在选中编辑器窗口时按左右箭头能后退或前进一帧去预览。

动作轨道

在动作轨道上没有 clip 的帧里按**右键**，能弹出当前预览 Animator 里用到的所有 AnimationClip 以供选择，也可以从 Project 窗口里**拖**一个 AnimationClip 资源到该轨道里，这样都能新建一个 clip。动作轨道在游戏 runtime 的时候实际驱动的也是 Animator 组件。clip 在轨道中显示 AnimationClip 的名字。

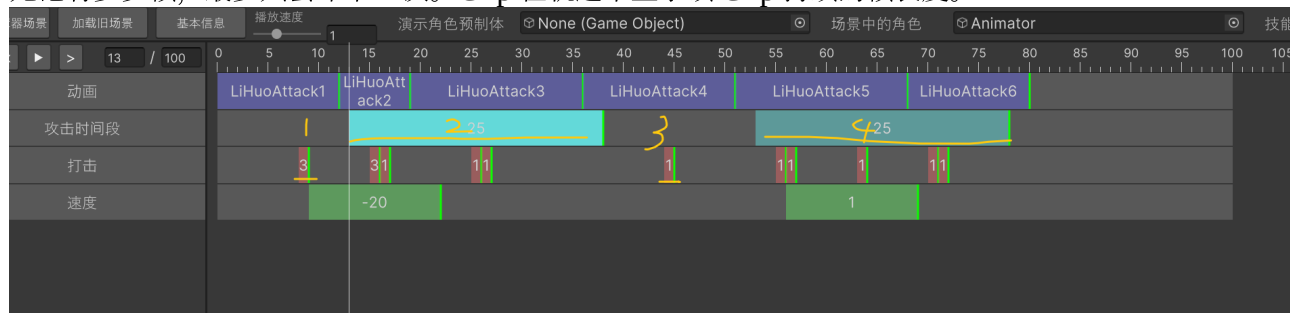


速度轨道

在速度轨道上没有 clip 的帧里按**右键**，能直接新建一段 clip。Clip 在轨道中显示 Vector2 的速度，如果速度的 y 分量近似为 0，则只显示 x 分量。速度轨道在游戏运行 runtime 中实际驱动的是 Rigidbody2D 组件。

攻击时间段轨道

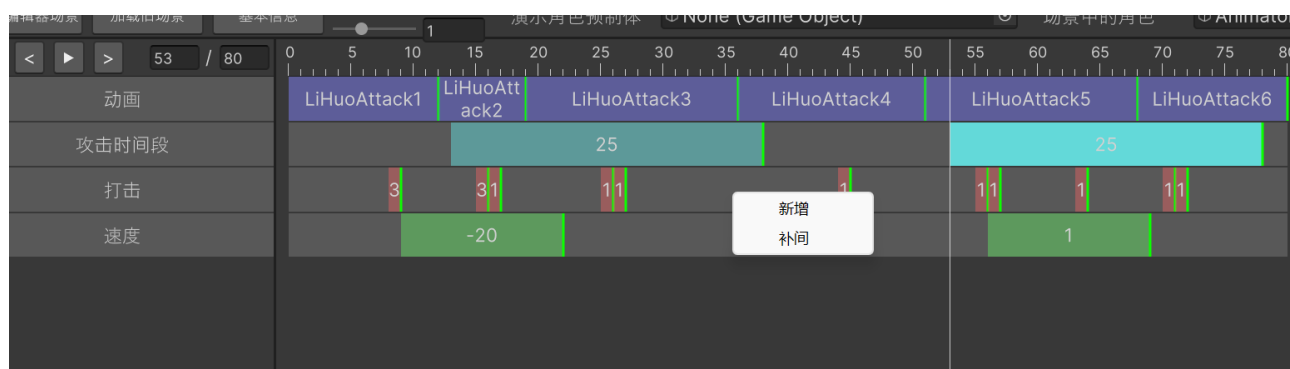
在攻击时间段轨道上没有 clip 的帧里按**右键**，能直接新建一段 clip，存在 clip 的帧数里，下方的 Hitbox 无论有多少帧，最多只会命中 1 次。Clip 在轨道中显示该 Clip 持续的帧长度。



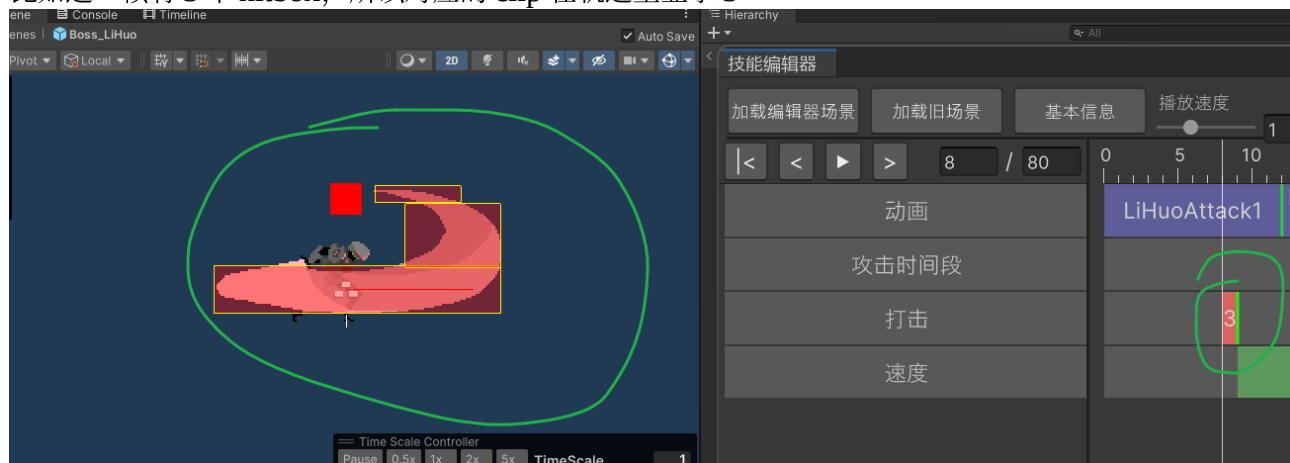
如图所示，这个技能有 11 个帧里存在 hitbox 判定，如果攻击时间段轨道里没有内容，那这个技能最多能打击中 11 次，但是现在因为攻击时间段轨道里存在内容，那最多只能打击中 4 次，分别是第 8 帧（橙 1），第 13-37 帧（橙 2），第 44 帧（橙 3），第 53-77 帧（橙 4）。

打击

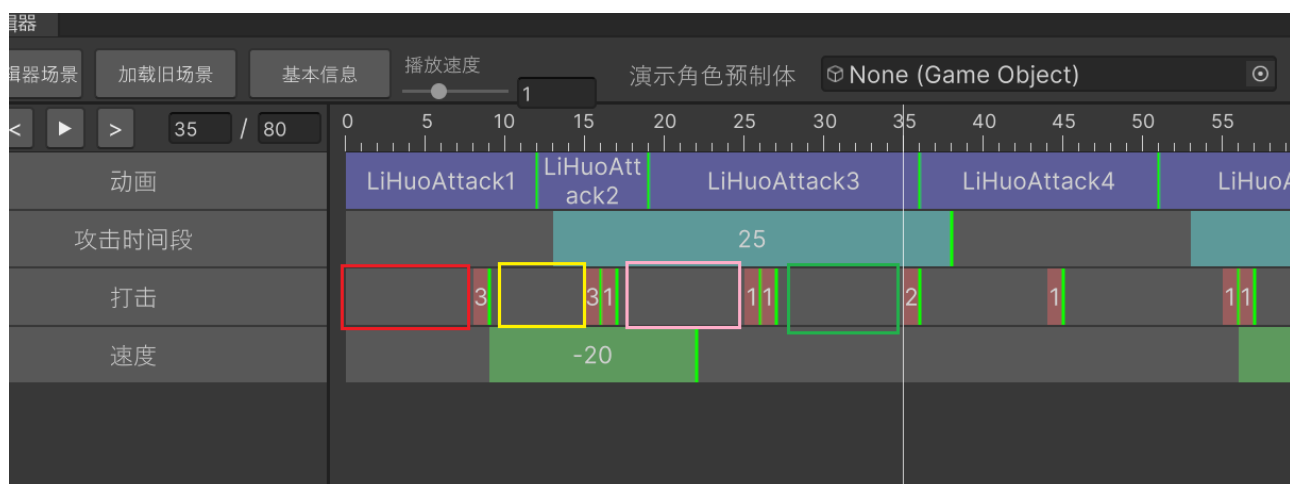
在打击轨道上没有 clip 的地方按**右键**，能选择是新建一个 clip 还是补间生成一段连续的 clip。Clip 在轨道中显示该 Clip 里 HitBox 的个数。



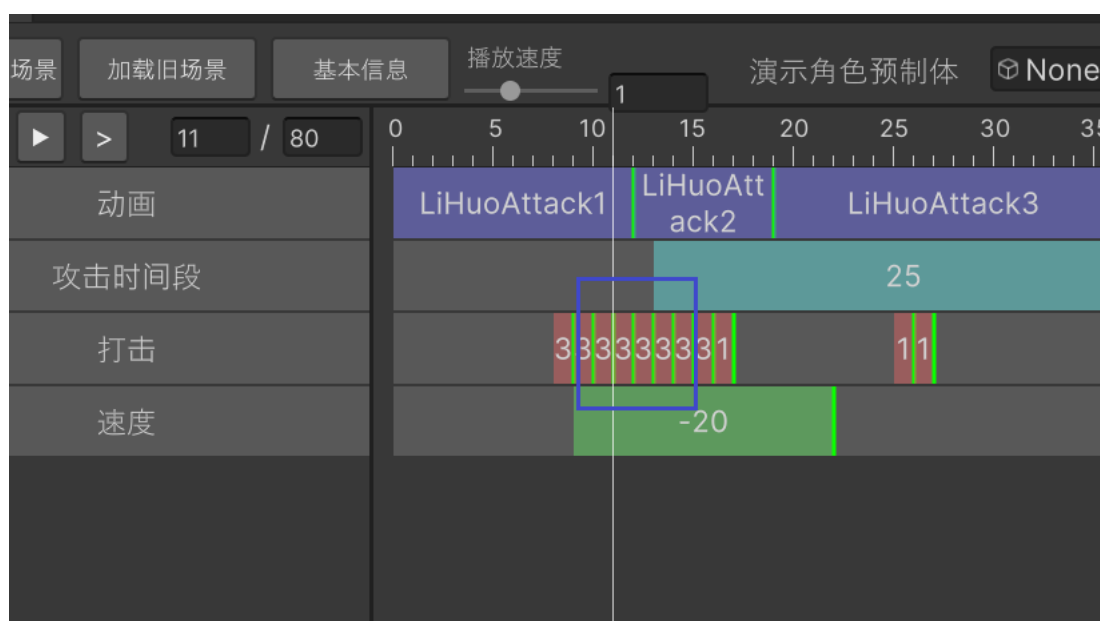
比如这一帧有 3 个 hitbox，所以对应的 clip 在轨道里显示 3



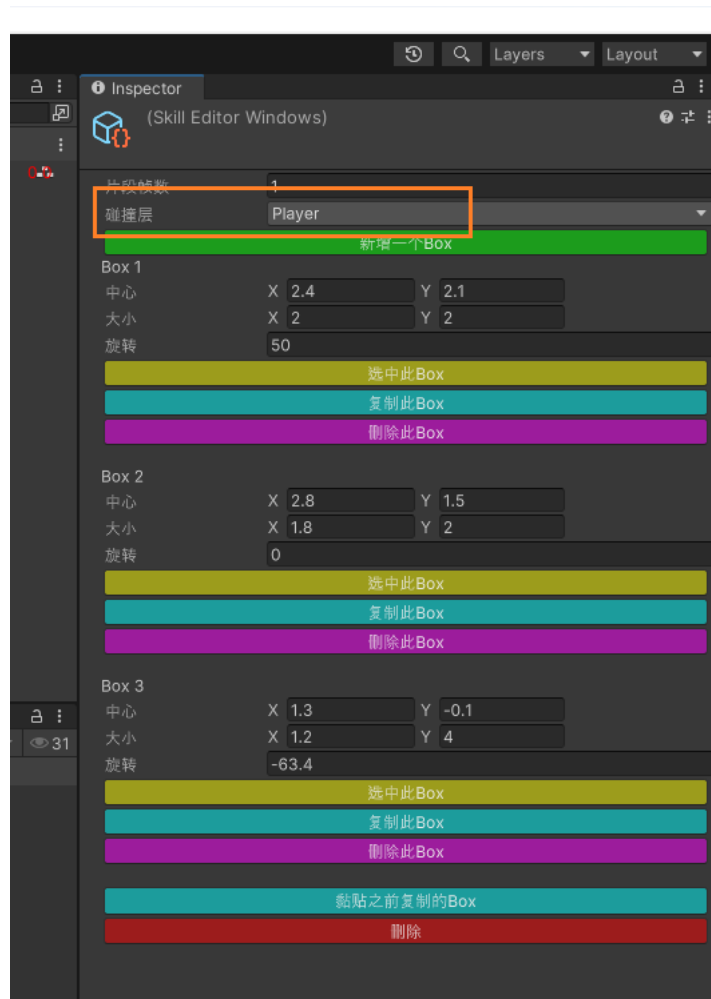
hitbox 补间功能使用的条件是该帧的左边能找到一个 clip，右边也能找到一段 clip，并且左右两边的 clip 里 hitbox 盒子数相等且非 0。



如图，红色段和绿色段无法补间，黄色段和粉色段可以补间。



补间后会对左右两端的所有 hitbox 盒子进行中心/大小/旋转进行线性 lerp 插值，并补充到逐个帧里。这样可以方便制作类似刀光的技能的 hitbox，可以在第一帧和最后一帧拉好对应的 hitbox，然后进行补间，这样就不用每帧都去手拉 hitbox 了。



打击的 clip 要指定对应碰撞的 **layer**，实际在游戏的运行 runtime 时会基于这个 layer 去做 Physics2D.OverlapBoxAll。

如何使用技能编辑器编辑后的 skill 资源（程序相关）

参考 `GamePlayEnemy.cs` 需要在用之前 new 一个 `SkillDriver` 对象，然后使用技能时需要调用 `SkillDriver` 的 `SetSkill` 方法，之后执行 `PlayAsync` 或者 `PlayFromFram` 方法，这两个方法都是异步的，会在技能执行完成后有对应的 `OnSkillFinish`。 `SkillDriver` 的 `SetSkill` 需要一个 `SkillConfig` 参数，也就是技能编辑器编辑的资源。加载一个技能资源需要 `var skill = await`

`ResourceManager.LoadAssetAsync<SkillConfig>(skillPath.value, ResType.ScriptObject);` 类似这样的异步加载过程，但是使用 `ResourceManager` 之前需要现在代码里执行 `await ResourceManager.Init();` 这个初始化。

目前技能编辑器和技能的 runtime 驱动支持时间的缩放和暂停。

未来会提供更多的轨道，比如投射物轨道。